


# Penerapan Grey Wolf Optimizer dalam Pelatihan Multi Layer Perceptron untuk Menangani Masalah Klasifikasi dan Regresi

Azminuddin I. S. Azis<sup>1</sup>, Budy Santoso<sup>2</sup>, Jeffrey<sup>3</sup>

Prodi Ilmu Komputer, Institut Teknologi Bacharuddin Jusuf Habibie<sup>1,3</sup>  
Prodi Teknik Komputer dan Jaringan, Politeknik Negeri Ujung Pandang<sup>2</sup>  
Email Korespondensi Author: [azminuddinazis@ith.ac.id](mailto:azminuddinazis@ith.ac.id)

This is an open access article under the [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/) license. 

## Kata kunci:

Multi Layer Perceptron, Grey Wolf Optimizer, Artificial Neural Network, Klasifikasi, Regresi.

## Abstrak

Algoritma *Grey Wolf Optimizer* (GWO) merupakan salah satu metode metaheuristik terkini yang telah terbukti mampu menunjukkan kinerja yang handal dalam memecahkan berbagai masalah optimasi, terutama dalam mengoptimalkan parameter pada algoritma-algoritma *Machine Learning*. Salah satu algoritma *Machine Learning* yang populer adalah *Multi Layer Perceptron* (MLP), merupakan salah satu varian dari *Artificial Neural Network* (ANN) yang memiliki parameter *weight* dan *bias* yang sensitif terhadap kinerja modelnya. Oleh karenanya, GWO diterapkan untuk mengoptimalkan inisialisasi awal *weight* dan *bias* dalam pelatihan MLP untuk meningkatkan kinerja modelnya. Hasil eksperimen ini menunjukkan bahwa optimalisasi GWO mampu meningkatkan kinerja MLP, baik pada klasifikasi Iris yang akurasi meningkat sebesar 33.33% dan pada regresi Silica dengan RMSE yang menurun sebesar 0.1488.

## Keywords:

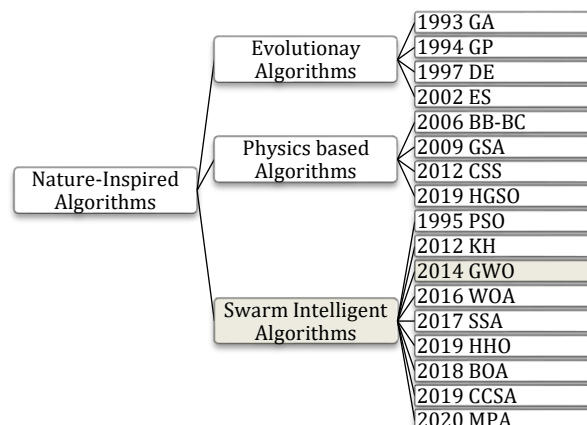
Multi Layer Perceptron, Grey Wolf Optimizer, Artificial Neural Network, Classification, Regression.

## Abstrack

*The Grey Wolf Optimizer (GWO) algorithm is one of the latest metaheuristic methods that has been proven to be able to show reliable performance in solving various optimization problems, especially in optimizing parameters in Machine Learning algorithms. One of the popular Machine Learning algorithms is the Multi Layer Perceptron (MLP), which is a variant of the Artificial Neural Network (ANN) that has weight and bias parameters that are sensitive to the performance of its model. Therefore, GWO is applied to optimize the initial initialization of weight and bias in MLP training to improve model performance. The results of this experiment show that GWO optimization is able to improve MLP performance, both in Iris classification whose accuracy increased by 33.33% and in Silica regression with RMSE decreasing by 0.1488.*

## Pendahuluan

Sebagian besar algoritma-algoritma metaheuristik terinspirasi oleh alam (*Nature-Inspired Algorithms*), meniru konsep Biologis dan Fisika, yang dapat dibagi menjadi tiga kategori, yaitu *Evolutionary Algorithms*, *Physics-based Algorithms*, dan *Swarm Intelligent Algorithms* (Banaie-Dezfouli et al., 2021) sebagaimana ditunjukkan pada Gambar 1. Hingga saat ini, algoritma metaheuristik yang berbeda telah diterapkan untuk memecahkan berbagai masalah optimasi. Salah satunya sering digunakan untuk memecahkan masalah-masalah optimasi pada algoritma-algoritma *Machine Learning*.



Gambar 1. Klasifikasi Algoritma-Algoritma Metaheuristik

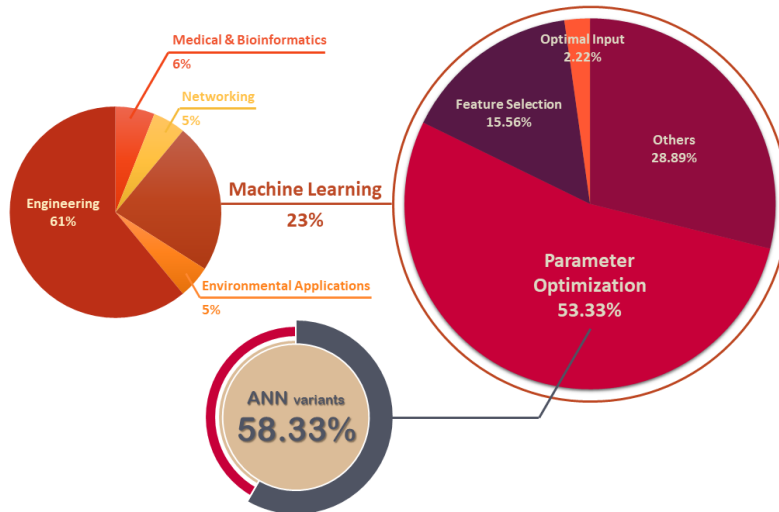
Algoritma *Grey Wolf Optimizer (GWO)* merupakan salah satu dari *Nature-Inspired Algorithms* pada kategori *Swarm Intelligent Algorithms* yang populer (Mirjalili et al., 2014), karena telah terbukti menunjukkan kinerja yang lebih handal dibandingkan *Swarm Intelligent Algorithms* lainnya dalam memecahkan berbagai masalah optimasi (Banaie-Dezfouli et al., 2021), (Faris et al., 2017). Mirjalili, dkk. memperkenalkan GWO yang terinspirasi oleh perilaku berburu dan kepemimpinan sosial *Grey Wolf* (Mirjalili et al., 2014). Kelebihan algoritma ini adalah sederhana, mudah digunakan, parameter yang sangat sedikit, fleksibel, dan terukur (Faris et al., 2017), (Long et al., 2018) sehingga sering diterapkan dalam menyelesaikan berbagai masalah optimasi dalam berbagai bidang, terutama dalam bidang Teknik, Medis, Bioinformatika, Jaringan, lingkungan aplikasi, dan juga *Machine Learning*. Dalam bidang *Machine Learning*, GWO biasanya digunakan untuk menyelesaikan masalah *parameter optimization*, *feature selection*, *optimal input*, dan lainnya sebagaimana ditunjukkan pada Gambar 2. Secara rinci, penerapan GWO dapat ditunjukkan pada Tabel 1 berikut ini.

Tabel 1. *State-of-the-art* Penerapan Algoritma GWO

Referensi	Metode yang Diusulkan	Topik	Tujuan
(Muangkote et al., 2014)	IGWO – q-Gaussian Radial Basis Functional-Link NN	Classification	Parameter optimization
(Gholizadeh, 2015)	Sequential GWO	Civil Engineering	Others optimization
(Komaki & Kayvanfar, 2015)	GWO	Scheduling problem	Others optimization
(Mirjalili, 2015)	GWO – ANN	Classification	Parameter optimization
(Jayapriya & Arock, 2015)	GWO	Bioinformatics	Others optimization
(Korayem et al., 2015)	K-GWO (K-Means GWO)	Clustering & routing	Parameter optimization
(Emary et al., 2015)	GWO	General	Feature selection
(Elhariri et al., 2015)	GWO – SVM	Image processing	Parameter optimization
(Eswaramoorthy et al., 2016)	GWO – SVM	Classification	Parameter optimization
(Hassanin et al., 2016)	GWO – Back Propagation	Classification	Parameter optimization
(Turabieh, 2016)	GWO – ANN	Regression	Parameter optimization
(L. Li et al., 2016)	FGWOA (Fuzzy based GWO Aggregation)	Image processing	Parameter optimization
(Saxena & Kothari, 2016)	GWO	Pattern synthesis	Others optimization
(Kumar et al., 2016)	GWO	System reliability	Others optimization
(Sodeifian et al., 2016)	GWO	Chemical	Others optimization
(Emary et al., 2016)	Binary GWO	General	Feature selection
(Bian et al., 2017)	GWO – SVM	Regression	Parameter optimization
(Precup et al., 2017)	GWO – Fuzzy Takagi-Sugeno	General	Parameter optimization
(Rajakumar et al., 2017)	GWO – Localization in Wireless Sensor Networks	Node localization	Others optimization
(Sanchez et al., 2017)	GWO – Modular Gramular Neural Networks	Image processing	Parameter optimization
(Wei et al., 2017)	IGWO – SVM	Classification	Parameter optimization
(Yassien et al., 2017)	GWO – 0/1 Knapsack Problem	Knapsack problem	Others optimization
(Zhang et al., 2017)	EOGWO (Elite Operation GWO)	General	Others optimization
(Q. Li et al., 2017)	IGWO – Kernel Extreme Learning Machine	Medical diagnosis	Others optimization
(Tsai et al., 2017)	MOGWO (Multi Objective GWO)	Robot path planning	Others optimization
(Kumaran et al., 2018)	GWO – CNN	Image processing	Parameter optimization
(Djema et al., 2019)	GWO – ANN	Regression	Parameter optimization
(Tikhmarine et al., 2019)	GWO – ANN	Regression	Parameter optimization
(Ardabili et al., 2020)	GWO – ANN	Regression	Parameter optimization
(Chantar et al., 2020)	GWO – Binary SVM	Text mining	Feature selection
(El-Kenawy et al., 2020)	KNN – Modified binary GWO – Stochastic Fractal Search	General	Feature selection
(Golafshani et al., 2020)	ANN – ANFIS – GWO	Regression	Parameter optimization
(Gupta & Deep, 2020)	Memory-based GWO	Image processing	Others optimization
(Hu et al., 2020)	KNN – Binary GWO	General	Feature selection
(X. Li & Luk, 2020)	GWO	Electromagnetics	Others optimization
(Majumder & Eldho, 2020)	GWO – ANN	Regression	Parameter optimization
(Nosratabadi et al., 2020)	GWO – ANN	Regression	Parameter optimization
(Shariati et al., 2020)	GWO – Extreme Learning Machine	Regression	Parameter optimization
(Tikhmarine et al., 2020)	GWO – SVM	Regression	Parameter optimization
(Maroufpoor et al., 2020)	GWO – ANN	Regression	Optimal input
(Xie et al., 2020)	GWO based CNN-LSTM	Regression	Parameter optimization
(Gayathri & Joseph Jawhar, 2021)	GWO -CNN	Image processing	Feature selection
(Liu et al., 2021)	Enhanced-GWO – SVR	Regression	Feature selection
(Uzlu, 2021)	GWO – ANN	Regression	Parameter optimization
(Reddy & Gurralla, 2022)	Convulotional based GWO with Variable Weights	Classification	Parameter optimization

Implementasi GWO dalam bidang *Machine Learning* dari tahun 2014 (tahun diperkenalkannya GWO) hingga saat ini paling sering digunakan untuk *parameter optimization*, secara statistik dapat ditunjukkan pada Gambar 2. Sebagaimana diketahui bahwa memang setiap algoritma *Machine Learning* memiliki satu atau lebih parameternya yang sensitif terhadap kinerja modelnya, sehingga optimalisasi terhadap parameter-parameter yang sensitif tersebut akan dapat meningkatkan kinerja model *Machine*

*Learning*. Misalnya seperti parameter *weight* dan *bias* pada algoritma *Artificial Neural Network* (ANN) (Mirjalili, 2015), paling sering mendapatkan perhatian sebagai objek penerapan optimalisasi GWO.

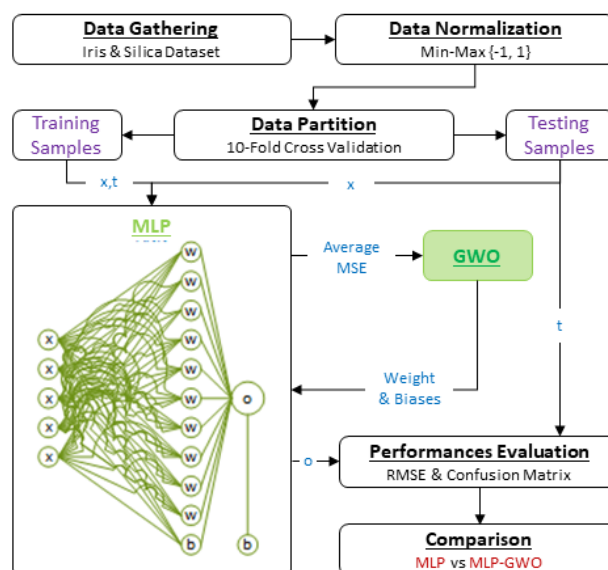


Gambar 2. Statistik Penerapan GWO untuk Optimalisasi (Sumber: Tabel 1 & (Faris et al., 2017))

ANN merupakan salah satu algoritma populer *Machine Learning*, sedangkan Algoritma *Multi Layer Perceptron* (MLP) atau *Feed Forward Network* (FFN) merupakan varian ANN yang paling umum yang dapat digunakan untuk menyelesaikan masalah estimasi/regresi, klasifikasi, dll (Santoso et al., 2020). Salah satu kelemahan MLP terdapat pada parameter *weight* dan *bias* yang sensitif terhadap kinerja model (Mirjalili, 2015). Maksudnya, karena penentuan inisialisasi nilai *weight* dan *bias* dilakukan secara acak, maka masalah tersebut akan sangat mempengaruhi kinerja model MLP. Untuk mengatasi masalah tersebut, optimalisasi terhadap inisialisasi *weight* dan *bias* dalam pelatihan model MLP dapat menjadi solusi alternatif yang tepat. Oleh karenanya, GWO yang handal dalam menyelesaikan masalah optimalisasi dapat diterapkan untuk menangani masalah MLP tersebut. Inilah yang menjadi salah satu motivasi kami dalam penelitian ini, menerapkan GWO untuk mengoptimalkan inisialisasi parameter *weight* dan *bias* dalam pelatihan model MLP sehingga dapat meningkatkan kinerja model MLP dalam menangani masalah klasifikasi dan regresi.

## Metode

Secara umum, metode yang diusulkan atau tahapan penelitian ini dimulai dari pengumpulan data, normalisasi data, partisi/validasi data, pelatihan model, pengujian/evaluasi model, dan komparasi model. Lebih jelasnya, dapat ditunjukkan pada Gambar 3 berikut ini.



Gambar 3. Metode yang Diusulkan

## 1. Dataset

Pengumpulan data dilakukan dengan menggunakan teknik dokumentasi, yaitu dataset publik Iris dan Silica yang dikumpulkan dari *UCI Machine Learning Repository*. Adapun penjelasan ringkas dari kedua dataset tersebut dapat ditunjukkan pada Tabel 2 berikut ini.

Tabel 2. Deskripsi Dataset

Kode	Nama Dataset	Jumlah Instances	Jumlah Attribut	Missing Value	Target
D1	Iris	150	4	No	3 Label Class
D2	Silica	2339	3	No	Regression

## 2. Min-Max Normalization and Data Validation

Normalisasi data merupakan proses yang sangat berguna untuk MLP ketika dataset yang memiliki fitur-fitur dengan rentang yang berbeda-beda. Oleh sebab itu, pada prosedur pra-pengolahan data, diterapkan pendekatan *data normalization* menggunakan metode *Min-Max Normalization* dalam rentang  $\{-1, 1\}$ , yang mana persamaannya dapat didefinisikan sebagai berikut.

$$y = \frac{(y_{\max} - y_{\min})(x - x_{\min})}{(x_{\max} - x_{\min})} + y_{\min} \quad (1)$$

$y$  adalah hasil normalisasi dari  $x$ ,  $y_{\max}$  dan  $y_{\min}$  adalah nilai maksimal dan minimal dari data,  $x_{\max}$  dan  $x_{\min}$  adalah nilai range maksimal dan minimal untuk normalisasi.

Sedangkan untuk partisi maupun validasi data menggunakan pendekatan *10-Fold Cross Validation*, sehingga akan terdapat 10% data uji yang berbeda-beda di setiap pelatihan dan pengujian model yang dilakukan sebanyak 10 kali.

## 3. Grey-Wolf Optimizer

Dalam algoritma GWO, populasi awal terdiri dari empat jenis *Grey Wolf*, yaitu *alfa* ( $\alpha$ ), yang merupakan solusi terbaik pertama, sehingga solusi terbaik kedua dan ketiga disebut *beta* ( $\beta$ ) dan *delta* ( $\delta$ ), serta setiap individu dianggap sebagai *omega* ( $\omega$ ) (Mirjalili et al., 2014), (Banaie-Dezfouli et al., 2021). Terdapat tiga langkah utama, yaitu melingkari, berburu, dan menyerang (Mirjalili et al., 2014), (Banaie-Dezfouli et al., 2021). Perilaku melingkari dapat dimodelkan pada Persamaan (2) dan (3) berikut ini.

$$D_i = |c_i x_p(t) - x_i(t)| \quad (2)$$

$$x_i(t+1) = x_p(t) - A_i D_i \quad (3)$$

Posisi *Grey Wolf* ke- $i$  adalah iterasi saat ini, sedangkan posisi mangsa adalah  $x_i, t, x_p$ . Vektor  $A$  dan  $C$  adalah parameter kontrol yang dihitung dengan Persamaan (4) dan (5) berikut ini.

$$A_i = 2 \cdot a \cdot r_1 - a \quad (4)$$

$$C_i = 2 \cdot r_2 \quad (5)$$

Vektor  $r_1$  dan  $r_2$  adalah nilai acak dari  $[0, 1]$ , sedangkan  $a$  menurun secara linier dari 2 ke 0 dalam setiap iterasi yang dihitung dengan Persamaan (6) berikut ini.

$$a = 2 - 2 \cdot t / t_{\max} \quad (6)$$

Pemimpin Serigala yang memperkirakan kemungkinan posisi mangsa, sehingga seluruh *omega* dipandu oleh pemimpin (*alfa*, *beta*, dan *delta*). Posisi setiap *omega*  $x_i$  diperbaharui dengan menggunakan Persamaan (7)-(13) berikut ini.

$$D_\alpha = |C_1 x_\alpha(t) - x_i(t)| \quad (7)$$

$$D_\beta = |C_2 x_\beta(t) - x_i(t)| \quad (8)$$

$$D_\delta = |C_3 x_\delta(t) - x_i(t)| \quad (9)$$

$$x_1 = x_\alpha(t) - A_1 D_\alpha \quad (10)$$

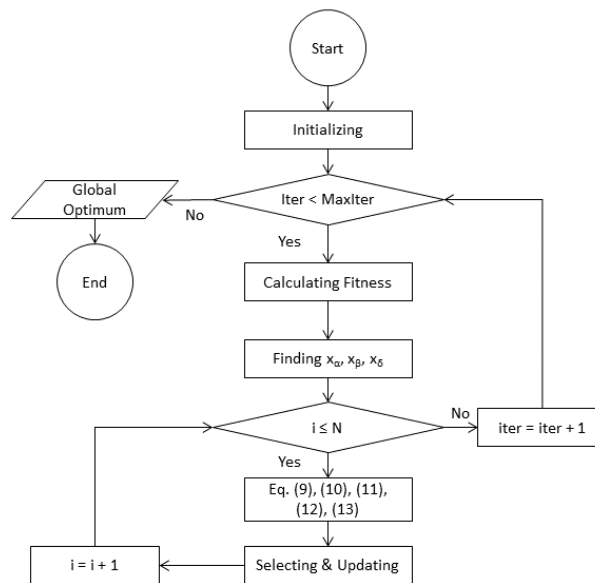
$$x_2 = x_\beta(t) - A_1 D_\beta \quad (11)$$

$$x_3 = x_\delta(t) - A_1 D_\delta \quad (12)$$

$$x_i(t+1) = (x_1 + x_2 + x_3)/3 \quad (13)$$

$x_\alpha, x_\beta$ , dan  $x_\delta$  menunjukkan posisi *alfa*, *beta*, dan *delta*, sedangkan  $x_i$  menunjukkan posisi *omega* ke- $i$ .

Adapun algoritma GWO dapat ditunjukkan pada Gambar 4 berikut ini.



Gambar 4. Algoritma GWO

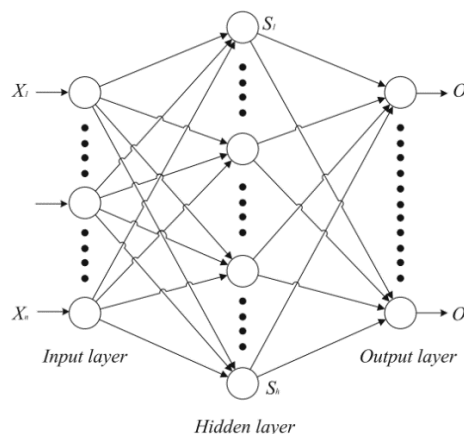
#### 4. Multi Layer Perceptron

MLP merupakan varian yang paling umum dari ANN yang merupakan salah satu algoritma *Machine Learning* yang dapat digunakan untuk estimasi/regresi, klasifikasi, dll (Santoso et al., 2020). ANN bekerja meniru cara kerja otak manusia dari sisi: (1) pengetahuan yang diperoleh oleh jaringan dari lingkungan, melalui suatu proses pembelajaran; (2) kekuatan koneksi antar unit yang disebut *synaptic weights*, berfungsi untuk menyimpan pengetahuan yang telah diperoleh oleh jaringan tersebut (Santoso et al., 2020). Pada tahun 1943, Mc. Culloch dan Pitts memperkenalkan model matematika yang merupakan penyederhanaan dari struktur sel saraf yang sebenarnya yang ditunjukkan pada Persamaan (14) berikut ini.

$$y = f\left(\sum_{i=1}^n x_i w_i\right) \quad (14)$$

Korelasi antara ketiga komponen pada persamaan di atas yaitu: signal ( $x$ ) berupa vektor berdimensi  $n$  ( $x_1, x_2, \dots, x_n$ ) akan mengalami penguatan oleh synapse  $w$  ( $w_1, w_2, \dots, w_n$ ). Selanjutnya akumulasi dari penguatan tersebut akan mengalami transformasi oleh fungsi aktivasi  $f$ . Fungsi  $f$  ini akan memonitor, bila akumulasi penguatan signal itu telah melebihi batas tertentu.

Pada MLP, ANN terdiri dari beberapa layer. Layer pertama adalah input, layer terakhir adalah output, dan layer diantara keduanya disebut hidden layer  $\geq 1$  (Mirjalili, 2015). Suatu arsitektur jaringan ANN dengan satu hidden layer dapat diilustrasikan seperti pada Gambar 5 berikut ini.



Gambar 5. Jaringan ANN (MLP) dengan 1 *Hidden Layer* (Mirjalili et al., 2014)

## 5. Classification Performances

Suatu model klasifikasi dapat diukur kinerjanya berdasarkan tabel *Confusion Matrix*, sebagaimana ditunjukkan pada Tabel 3 berikut ini.

Tabel 3. *Confusion Matrix*

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Positive (FN)
Actual Negative	False Negative (FP)	True Negative (TN)

Berdasarkan tabel *Confusion Matrix*, maka kinerja akurasi dari suatu model klasifikasi dapat diukur menggunakan Persamaan (15) berikut ini.

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FN} + \text{FP})} \quad (15)$$

## 6. Regression Performances

Suatu model regresi/estimasi dapat diukur kinerjanya dengan menggunakan pendekatan *error estimation*, seperti *Root Mean Squared Error* (RMSE) yang didefinisikan pada Persamaan (16) berikut.

$$\text{RMSE} = \sqrt{\sum_{i=1}^n (y_i - y'_i)^2 / n} \quad (16)$$

Dimana  $n$  adalah jumlah data,  $y$  adalah target pada data ke- $i$ , dan  $y'$  adalah *output* yang diprediksi oleh model pada data ke- $i$ .

Sedangkan untuk mengukur kinerja korelasi antara *target* (*actual output*) dan *output* (*output* yang diprediksi model) dari suatu model regresi/estimasi menggunakan pendekatan  $R^2$  yang didefinisikan pada Persamaan (17) berikut ini.

$$R^2 = \frac{\sum_{i=1}^n (X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum_{i=1}^n (X - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y - \bar{Y})^2}} \quad (17)$$

Dimana  $X$  adalah target,  $\bar{X}$  = rata-rata  $X$ ,  $Y$  adalah output yang diprediksi,  $\bar{Y}$  adalah rata-rata dari  $Y$ , dan  $n$  adalah jumlah data.

## Hasil dan Diskusi

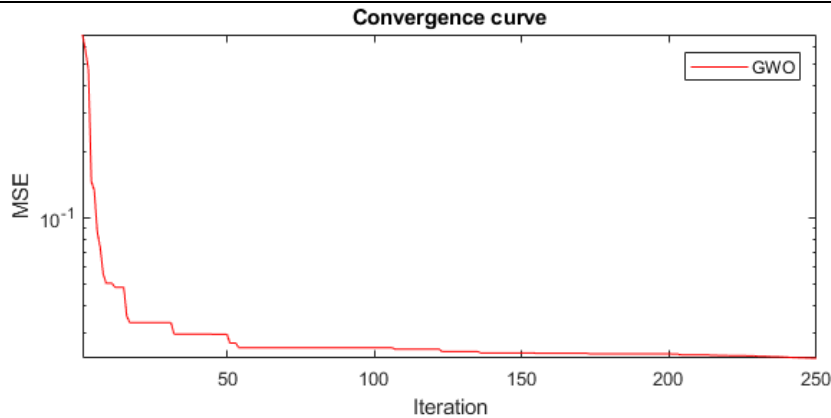
### 1. Klasifikasi Iris

Penerapan GWO untuk mengoptimalkan inisialisasi parameter *weight* dan *bias* dalam pelatihan MLP dengan model parameter: *transfer function* = *logsig*, *iteration* = 250, *neurons size* = 11, *number of search agents* = 50, *dimensions* = 78, *lower bound* = -50, dan *upper bound* = 50 berhasil meningkatkan kinerja akurasi model MLP pada klasifikasi dataset Iris yang secara rinci dapat ditunjukkan pada Tabel 4 berikut ini.

Tabel 4. Komparasi Kinerja Akurasi Model MLP vs MLP-GWO pada Klasifikasi Iris

Model	Akurasi
MLP (Metode standar)	92.59%
MLP-GWO (Metode yang diusulkan)	59.26%

Dengan demikian penerapan GWO untuk mengoptimalkan parameter *weight* dan *bias* dalam pelatihan MLP berhasil meningkatkan kinerja akurasi model MLP pada klasifikasi Iris, yaitu sebesar 33.33%. Hasil ini diperoleh/diperkuat karena penurunan (perbaikan) nilai *Mean Squared Error* (MSE) dalam pelatihan MLP pada setiap iterasi proses GWO, dimana MSE pada iterasi ke-50 sebesar 0.029511 dan terus menurun hingga iterasi ke-250 sebesar 0.02297. Lebih jelasnya, dapat ditunjukkan pada *convergence curve* Gambar 6 berikut ini.



Gambar 6. *Convergence Curve* MLP-GWO pada Klasifikasi Iris

## 2. Regresi Silica

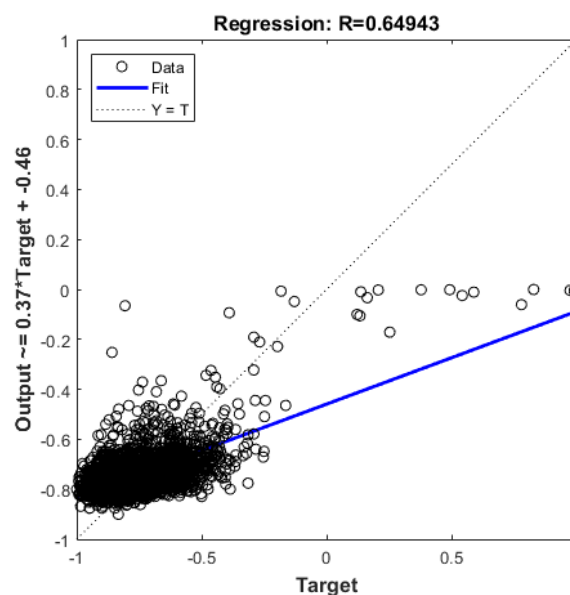
Penerapan GWO untuk mengoptimalkan inisialisasi parameter *weight* dan *bias* dalam pelatihan MLP dengan model parameter: *transfer function = logsig*, *iteration = 250*, *neurons size = 11*, *number of search agents = 50*, *dimensions = 78*, *lower bound = -50*, dan *upper bound = 50* berhasil meningkatkan kinerja *error estimation* (RMSE) model MLP pada regresi dataset Silica yang secara rinci dapat ditunjukkan pada Tabel 5 berikut ini.

Tabel 5. Komparasi Kinerja RMSE Model MLP vs MLP-GWO pada Regresi Silica

Model	RMSE
MLP (Metode standar)	0.2793
MLP-GWO (Metode yang diusulkan)	0.1305

Dengan demikian penerapan GWO untuk mengoptimalkan parameter *weight* dan *bias* dalam pelatihan MLP berhasil meningkatkan kinerja model MLP pada regresi Silica, yaitu dengan penurunan RMSE sebesar 0.1488.

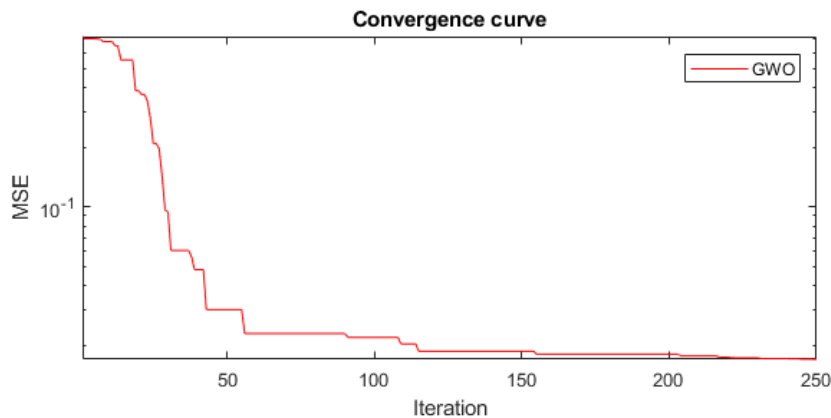
Sedangkan kurva *R-Regression* ( $R^2$ ) dari model MLP-GWO menunjukkan nilai sebesar 0.64943 yang mengindikasikan bahwa adanya cukup perbedaan antara *output* dan *target*, sebagaimana dapat ditunjukkan pada Gambar 7 berikut ini.



Gambar 7. *R-Regression* Model MLP-GWO pada Regresi Silica

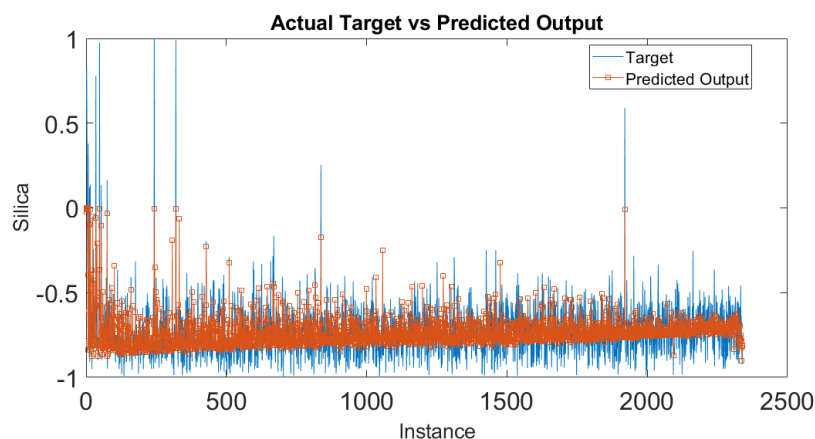
Hasil ini diperoleh/diperkuat karena penurunan (perbaikan) nilai *Mean Squared Error* (MSE) dalam pelatihan MLP pada setiap iterasi proses GWO, dimana MSE pada iterasi ke-10 sebesar 0.68114 dan

terus menurun hingga iterasi ke-250 sebesar 0.017022. Lebih jelasnya, dapat ditunjukkan pada *convergence curve* Gambar 8 berikut ini.



Gambar 8. *Convergence Curve* MLP-GWO pada Regresi Silica

Grafik komparasi antara *output* (*output* yang diprediksi model) dan *target* (*output* aktual) dari model MLP-GWO pada regresi Silica dapat ditunjukkan pada Gambar 9 berikut ini.



Gambar 9. *Target vs Output* dari Model MLP-GWO pada Regresi Silica

## Kesimpulan

Studi ini mengusulkan suatu model/metode MLP-GWO untuk menangani masalah klasifikasi dan regresi, dimana sampel dataset yang digunakan untuk klasifikasi adalah Iris, sedangkan untuk regresi adalah Silica. Pada klasifikasi Iris, model MLP-GWO menunjukkan akurasi sebesar 92.59%, meningkat 33.33% dibandingkan model MLP. Sedangkan pada regresi Silica, model MLP-GWO menunjukkan RMSE sebesar 0.1305, menurun 0.1488 dibandingkan model MLP. Hasil ini diperoleh/diperkuat karena penurunan (perbaikan) nilai MSE dalam pelatihan MLP pada setiap iterasi proses GWO, dimana iterasi terakhir memberikan nilai inialisasi parameter *weight* dan *bias* yang optimal untuk MLP. Kinerja model MLP-GWO tersebut diperoleh melalui model parameter: *transfer function* = *logsig*, *iteration* = 250, *neurons size* = 11, *number of search agents* = 50, *dimensions* = 78, *lower bound* = -50, dan *upper bound* = 50. Dengan demikian, penerapan GWO untuk mengoptimalkan inialisasi parameter *weight* dan *bias* dalam pelatihan MLP berhasil meningkatkan kinerja model MLP, baik pada masalah klasifikasi maupun regresi.

Perlu diketahui bahwa, GWO selalu dapat memberikan nilai *weight* dan *bias* di setiap iterasinya yang dapat menurunkan nilai MSE dalam pelatihan MLP, sehingga semakin banyak iterasi yang digunakan, maka semakin baik kinerja pelatihan MLP, namun tentu saja semakin membutuhkan *runtime* pula. Selain itu, pada masalah regresi Silica, kinerja *R-Regression* dari model MLP-GWO masih menunjukkan nilai yang belum memuaskan yang mengindikasikan target dan output yang belum rapat. Kedua hal ini dapat menjadi bahan untuk penelitian terkait selanjutnya.



## Referensi

- Ardabili, S., Mosavi, A., Band, S. S., & Varkonyi-Koczy, A. R. (2020). Coronavirus Disease (COVID-19) Global Prediction Using Hybrid Artificial Intelligence Method of ANN Trained with Grey Wolf Optimizer. *CANDO-EPE 2020 - Proceedings, IEEE 3rd International Conference and Workshop in Obuda on Electrical and Power Engineering, 2020*, 251–254. <https://doi.org/10.1109/CANDO-EPE51100.2020.9337757>
- Banaie-Dezfouli, M., Nadimi-Shahraki, M. H., & Beheshti, Z. (2021). R-GWO: Representative-based grey wolf optimizer for solving engineering problems. *Applied Soft Computing, 106*, 107328. <https://doi.org/10.1016/j.asoc.2021.107328>
- Bian, X. Q., Zhang, Q., Zhang, L., & Chen, J. (2017). A grey wolf optimizer-based support vector machine for the solubility of aromatic compounds in supercritical carbon dioxide. *Chemical Engineering Research and Design, 123*, 284–294. <https://doi.org/10.1016/j.cherd.2017.05.008>
- Chantar, H., Mafarja, M., Alsawalqah, H., Heidari, A. A., Aljarah, I., & Faris, H. (2020). Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification. *Neural Computing and Applications, 32*(16), 12201–12220. <https://doi.org/10.1007/s00521-019-04368-6>
- Djema, M. A., Boudour, M., Agbossou, K., Cardenas, A., & Doumbia, M. L. (2019). Adaptive direct power control based on ANN-GWO for grid interactive renewable energy systems with an improved synchronization technique. *International Transactions on Electrical Energy Systems, 29*(3), 1–15. <https://doi.org/10.1002/etep.2766>
- El-Kenawy, E. S. M., Eid, M. M., Saber, M., & Ibrahim, A. (2020). MbGWO-SFS: Modified Binary Grey Wolf Optimizer Based on Stochastic Fractal Search for Feature Selection. *IEEE Access, 8*, 107635–107649. <https://doi.org/10.1109/ACCESS.2020.3001151>
- Elhariri, E., El-Bendary, N., Hassanien, A. E., & Abraham, A. (2015). Grey Wolf Optimization for One-Against-One Multi-Class Support Vector Machines. *Proceedings of the 2015 7th International Conference of Soft Computing and Pattern Recognition, SoCPaR 2015*, 7–12. <https://doi.org/10.1109/SOCPAR.2015.7492781>
- Emary, E., Zawbaa, H. M., Grosan, C., & Hassenian, A. E. (2015). Feature Subset Selection Approach by Gray-Wolf Optimization. *Afro-European Conference for Industrial Advancement, 334*, 1–13. <https://doi.org/10.1007/978-3-319-13572-4>
- Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). Binary Grey Wolf Optimization Approaches for Feature Selection. *Neurocomputing, 172*, 371–381. <https://doi.org/10.1016/j.neucom.2015.06.083>
- Eswaramoorthy, S., Sivakumaran, N., & Sekaran, S. (2016). Grey Wolf Optimization based parameter selection for Support Vector Machines. *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, 35*(5), 1513–1523. <https://doi.org/10.1108/COMPEL-09-2015-0337>
- Faris, H., Aljarah, I., Al-Betar, M. A., & Mirjalili, S. (2017). Grey wolf optimizer: a review of recent variants and applications. *Neural Computing and Applications, 30*(2), 413–435. <https://doi.org/10.1007/s00521-017-3272-5>
- Gayathri, S. G., & Joseph Jawhar, S. (2021). A Novel IR Analyzer Based Property Extraction for Segmented Branch Retinal Artery Occlusion and GWO-CNN Based Classification–An Ophthalmic Outcome. *IETE Journal of Research*. <https://doi.org/10.1080/03772063.2021.1886876>
- Gholizadeh, S. (2015). Optimal design of double layer grids considering nonlinear behaviour by sequential grey wolf algorithm. *IUST, 5*(4), 511–523.
- Golafshani, E. M., Behnood, A., & Arashpour, M. (2020). Predicting the compressive strength of normal and High-Performance Concretes using ANN and ANFIS hybridized with Grey Wolf Optimizer. *Construction and Building Materials, 232*, 117266. <https://doi.org/10.1016/j.conbuildmat.2019.117266>
- Gupta, S., & Deep, K. (2020). A memory-based Grey Wolf Optimizer for global optimization tasks. *Applied Soft Computing Journal, 93*, 106367. <https://doi.org/10.1016/j.asoc.2020.106367>
- Hassanin, M. F., Shoeb, A. M., & Hassanien, A. E. (2016). Grey Wolf Optimizer-based Back-propagation Neural Network Algorithm. *2016 12th International Computer Engineering Conference, ICENCO 2016: Boundless Smart Societies*, 213–218. <https://doi.org/10.1109/ICENCO.2016.7856471>

- Hu, P., Pan, J. S., & Chu, S. C. (2020). Improved Binary Grey Wolf Optimizer and Its application for feature selection. *Knowledge-Based Systems*, 195(xxxx), 105746. <https://doi.org/10.1016/j.knosys.2020.105746>
- Jayapriya, J., & Arock, M. (2015). A parallel GWO technique for aligning multiple molecular sequences. *2015 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2015*, 210–215. <https://doi.org/10.1109/ICACCI.2015.7275611>
- Komaki, G. M., & Kayvanfar, V. (2015). Grey Wolf Optimizer Algorithm for the Two-stage Assembly Flowshop Scheduling Problem with Release Time. *Journal of Computational Science*, 8, 109–120. <https://doi.org/10.1016/j.jocs.2015.03.011>
- Korayem, L., Khorsid, M., & Kassem, S. S. (2015). Using grey Wolf algorithm to solve the capacitated vehicle routing problem. *IOP Conference Series: Materials Science and Engineering*, 83(1). <https://doi.org/10.1088/1757-899X/83/1/012014>
- Kumar, A., Pant, S., & Ram, M. (2016). System Reliability Optimization Using Gray Wolf Optimizer Algorithm. *Quality and Reliability Engineering International*, 33(7), 1327–1335. <https://doi.org/10.1002/qre.2107>
- Kumaran, N., Vadivel, A., & Kumar, S. S. (2018). Recognition of human actions using CNN-GWO: a novel modeling of CNN for enhancement of classification performance. In *Multimedia Tools and Applications* (Vol. 77, Issue 18). Multimedia Tools and Applications. <https://doi.org/10.1007/s11042-017-5591-z>
- Li, L., Sun, L., Kang, W., Guo, J., Han, C., & Li, S. (2016). Fuzzy Multilevel Image Thresholding Based on Modified Discrete Grey Wolf Optimizer and Local Information Aggregation. *IEEE Access*, 4(c), 6438–6450. <https://doi.org/10.1109/ACCESS.2016.2613940>
- Li, Q., Chen, H., Huang, H., Zhao, X., Cai, Z., Tong, C., Liu, W., & Tian, X. (2017). An Enhanced Grey Wolf Optimization Based Machine for Medical Diagnosis. *Computational and Mathematical Methods in Medicine*, 2017, 1–16.
- Li, X., & Luk, K. M. (2020). The Grey Wolf Optimizer and Its Applications in Electromagnetics. *IEEE Transactions on Antennas and Propagation*, 68(3), 2186–2197. <https://doi.org/10.1109/TAP.2019.2938703>
- Liu, M., Luo, K., Zhang, J., & Chen, S. (2021). A stock selection algorithm hybridizing grey wolf optimizer and support vector regression. *Expert Systems with Applications*, 179(September 2020), 115078. <https://doi.org/10.1016/j.eswa.2021.115078>
- Long, W., Jiao, J., Liang, X., & Tang, M. (2018). An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. *Engineering Applications of Artificial Intelligence*, 68(October 2017), 63–80. <https://doi.org/10.1016/j.engappai.2017.10.024>
- Majumder, P., & Eldho, T. I. (2020). Artificial Neural Network and Grey Wolf Optimizer Based Surrogate Simulation-Optimization Model for Groundwater Remediation. *Water Resources Management*, 34(2), 763–783. <https://doi.org/10.1007/s11269-019-02472-9>
- Maroufpoor, S., Bozorg-Haddad, O., & Maroufpoor, E. (2020). Reference evapotranspiration estimating based on optimal input combination and hybrid artificial intelligent model: Hybridization of artificial neural network with grey wolf optimizer algorithm. *Journal of Hydrology*, 588(April), 125060. <https://doi.org/10.1016/j.jhydrol.2020.125060>
- Mirjalili, S. (2015). How effective is the Grey Wolf optimizer in training multi-layer perceptrons. *Applied Intelligence*, 43(1), 150–161. <https://doi.org/10.1007/s10489-014-0645-7>
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, 69, 46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- Muangkote, N., Sunat, K., & Chiewchanwattana, S. (2014). An Improved Grey Wolf optimizer for Training q-Gaussian Radial Basis Functional-link Nets. *2014 International Computer Science and Engineering Conference (ICSEC)*, 209–214. <https://doi.org/10.1109/ICSEC.2014.6978196>
- Nosratabadi, S., Szell, K., Beszedes, B., Imre, F., Ardabili, S., & Mosavi, A. (2020). Comparative Analysis of ANN-ICA and ANN-GWO for Crop Yield Prediction. *Proceedings - 2020 RIVF International Conference on Computing and Communication Technologies, RIVF 2020*. <https://doi.org/10.1109/RIVF48685.2020.9140786>
- Precup, R. E., David, R. C., Szedlak-Stinean, A. I., Petriu, E. M., & Dragan, F. (2017). An Easily Understandable Grey Wolf Optimizer and Its Application to Fuzzy Controller Tuning. *Algorithms*,

- 10(2), 1–15. <https://doi.org/10.3390/a10020068>
- Rajakumar, R., Amudhavel, J., Dhavachelvan, P., & Vengattaraman, T. (2017). GWO-LPWSN: Grey Wolf Optimization Algorithm for Node Localization Problem in Wireless Sensor Networks. *Journal of Computer Networks and Communications*, 2017. <https://doi.org/10.1155/2017/7348141>
- Reddy, V. P. C., & Gurralla, K. K. (2022). Joint DR-DME classification using deep learning-CNN based modified grey-wolf optimizer with variable weights. *Biomedical Signal Processing and Control*, 73. <https://doi.org/https://doi.org/10.1016/j.bspc.2021.103439>
- Sanchez, D., Melin, P., & Castillo, O. (2017). A Grey Wolf Optimizer for Modular Granular Neural Networks for Human Recognition. *Computational Intelligence and Neuroscience*, 2017. <https://doi.org/10.1155/2017/4180510>
- Santoso, B., Azis, A. I. S., & Zohrahayaty. (2020). *Machine Learning and Reasoning Fuzzy Logic*. Deepublish.
- Saxena, P., & Kothari, A. (2016). Optimal Pattern Synthesis of Linear Antenna Array Using Grey Wolf Optimization Algorithm. *International Journal of Antennas and Propagation*, 2016. <https://doi.org/10.1155/2016/1205970>
- Shariati, M., Mafipour, M. S., Ghahremani, B., Azarhomayun, F., Ahmadi, M., Trung, N. T., & Shariati, A. (2020). A novel hybrid extreme learning machine–grey wolf optimizer (ELM-GWO) model to predict compressive strength of concrete with partial replacements for cement. *Engineering with Computers*. <https://doi.org/10.1007/s00366-020-01081-0>
- Sodeifian, G., Saadati Ardestani, N., Sajadian, S. A., & Ghorbandoost, S. (2016). Application of supercritical carbon dioxide to extract essential oil from *Cleome coluteoides* Boiss: Experimental, response surface and grey wolf optimization methodology. *Journal of Supercritical Fluids*, 114, 55–63. <https://doi.org/10.1016/j.supflu.2016.04.006>
- Tikhmarine, Y., Malik, A., Kumar, A., Souag-Gamane, D., & Kisi, O. (2019). Estimation of monthly reference evapotranspiration using novel hybrid machine learning approaches. *Hydrological Sciences Journal*, 64(15), 1824–1842. <https://doi.org/10.1080/02626667.2019.1678750>
- Tikhmarine, Y., Souag-Gamane, D., Najah Ahmed, A., Kisi, O., & El-Shafie, A. (2020). Improving artificial intelligence models accuracy for monthly streamflow forecasting using grey Wolf optimization (GWO) algorithm. *Journal of Hydrology*, 582(October 2019), 124435. <https://doi.org/10.1016/j.jhydrol.2019.124435>
- Tsai, P. W., Nguyen, T. T., & Dao, T. K. (2017). Robot path planning optimization based on multiobjective grey wolf optimizer. *Advances in Intelligent Systems and Computing*, 536, 166–173. [https://doi.org/10.1007/978-3-319-48490-7\\_20](https://doi.org/10.1007/978-3-319-48490-7_20)
- Turabieh, H. (2016). A Hybrid ANN-GWO Algorithm for Prediction of Heart Disease. *American Journal of Operations Research*, 06(02), 136–146. <https://doi.org/10.4236/ajor.2016.62016>
- Uzlu, E. (2021). Estimates of greenhouse gas emission in Turkey with grey wolf optimizer algorithm-optimized artificial neural networks. *Neural Computing and Applications*, 1. <https://doi.org/10.1007/s00521-021-05980-1>
- Wei, Y., Ni, N., Liu, D., Chen, H., Wang, M., Li, Q., Cui, X., & Ye, H. (2017). An Improved Grey Wolf Optimization Strategy Enhanced SVM and Its Application in Predictiong the Second Major. *Mathematical Problems in Engineering*, 2017.
- Xie, H., Zhang, L., & Lim, C. P. (2020). Evolving CNN-LSTM Models for Time Series Prediction Using Enhanced Grey Wolf Optimizer. *IEEE Access*, 8, 161519–161541. <https://doi.org/10.1109/ACCESS.2020.3021527>
- Yassien, E., Masadeh, R., Alzaqebah, A., & Shaheen, A. (2017). Grey Wolf Optimization Applied to the 0/1 Knapsack Problem. *International Journal of Computer Applications*, 169(5), 11–15. <https://doi.org/10.5120/ijca2017914734>
- Zhang, S., Luo, Q., & Zhou, Y. (2017). Hybrid Grey Wolf Optimizer Using Elite Opposition-Based Learning Strategy and Simplex Method. *International Journal of Computational Intelligence and Applications*, 16(2), 1–37. <https://doi.org/10.1142/S1469026817500122>